

<h1>

Web Standards and Cascading Style Sheets

</h1>

by
Bryan Stephenson

Web Standards

A general term for the formal standards and other technical specifications that define and describe aspects of the World Wide Web. - Wikipedia

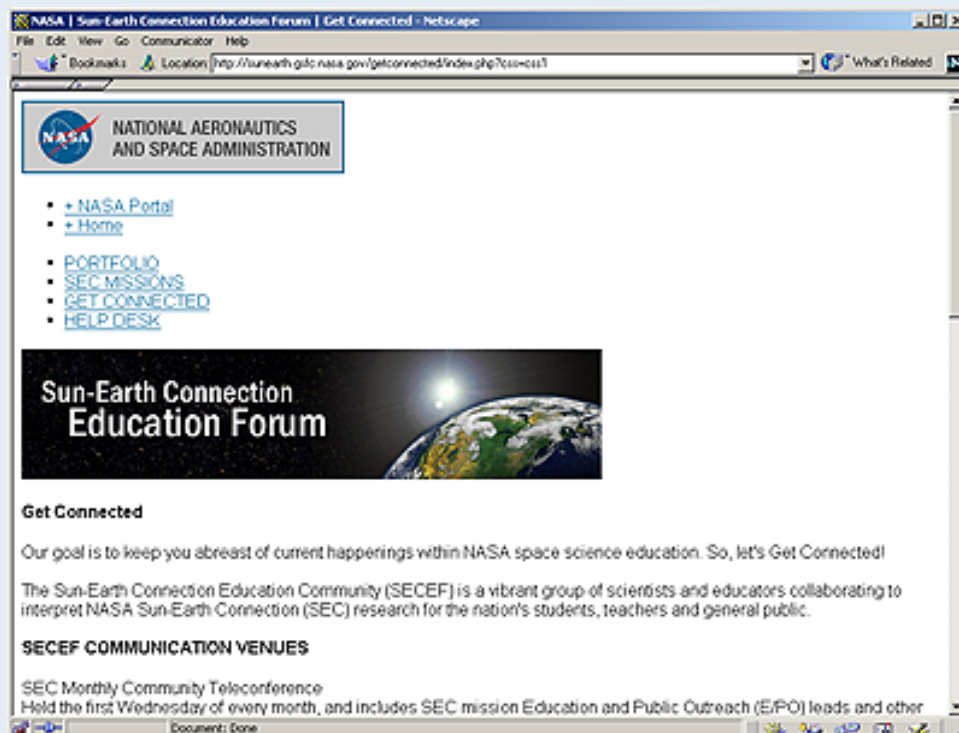
Cascading Style Sheets

A language used to control the presentation of another document such as an HTML or XHTML web page, or even an XML document.

A web page built with CSS but without following Web Standards will be a disaster. (but you could get lucky)

A web page built with Web Standards but without CSS will **NOT** be esthetically pleasing, or worse... boring. (but you could get lucky)

Without CSS



With CSS



Traditional vs Standards...

A different concept...

Traditional Table Web Sites:

Improper, messy, and hard to read mark-up.

Spacer gifs.

Non-logical document flow.

Requirements that the finished product look the same in certain browsers (backward compatibility).

Web Standards Web Sites:

Clean mark-up that is consistent and easy to read.

Extensive use of CSS for presentation.

Logical document flow.

Forward compatible strategy that is accessible in all web enabled browsers/devices.

Much easier to build, maintain, and re-design.

Traditional vs Standards...

Down the path to salvation...

Forget nearly everything you learned, especially the messy tricks used to make a page "look" right.

Take a deep breath...

Now say goodbye to tables, just let them go.

Content is key! Define the structure of your content without the presentation.

(X)HTML is very simple, so keep it that way. Every line of code should be clean and consistent.

Resist the temptation to misuse HTML to force a visual layout (I may be guilty of this).

Put accessibility ahead of esthetics.

Don't do it half-way. Don't try to mix traditional concepts with Web Standards. If you do, make sure you have a box of tissues and a bottle of Advil.

Be patient, there is a learning curve.

CSS, what is it good for?

Why CSS was a failure in the late 90's

I blame the browsers!.... Bill Gates is the devil!

The “Industry Norm” was not the “Industry Standard.”

Corporate hunger for web sites grew faster than the ability of browser makers and the W3C to satisfy their advertising needs.

Clients and managers decided that it was the web developer's job to compensate for the inability of browser makers (MS and Netscape) to develop a standards compliant product.

I would like to blame the browsers again.

With poor browser compliance, developers were more likely to take the easy way out.

Section 508c wasn't very scary back then.

If at first you don't succeed, quit and use tables.

2001: A Web Odyssey

We have come of age. Enter the Web Standards Project (WaSP).

Many web pioneers jump on the bandwagon to find fixes for many of the problems plaguing us.

Many web browser makers realize the catastrophe heading our way, and take the leap... except for our friends at Microsoft. (IE5 Macintosh developers get a special exception for actually trying, and being one of the first.)

Enter the age of CSS hacks, the short lived magic bullet to our problems.

Section 508c becomes a reality. Hold on to your carpal tunnel wrists, we're going for a ride!

2006, the coast is clear -

Browser developers learn to place nice in the sandbox, including Microsoft.

The popularity of Web 2.0 solidifies CSS's and Web Standards' place on the web.

Many popular web sites complete their switch to Web Standards and CSS, releasing the fear managers and clients have of it.

Hordes of developers stop in their tracks and totally change gears... The web God apologizes to the hordes of developers **"My bad, start over"*.

* there is no web God, and that was a satiricle quote

Changing gears... easy on that clutch...

If you have a tight deadline don't start from scratch. Make sure you have the time to do it properly.

Do your research before you begin. (my favorite resources are the the end of this document.)

Understanding the Box Model in CSS is imperative.

Make sure you have the right Doctype.

Beware the debates! Don't let a semantic zealot dissuade you on the right decision for your project.

The obvious: borrow code from fellow coders.

While initially developing your CSS always check your work in multiple browsers on multiple OS.

Don't be afraid to take chances with your CSS, but be more calculating when it comes to the structure of your HTML documents.

CSS and hacks - long live IE7

wait... what!?!.. OH NOOOO!

Hacking is a risky business; avoid it where possible.

The many box-model hacks. It's a difficult choice, and you'll likely have to use one.

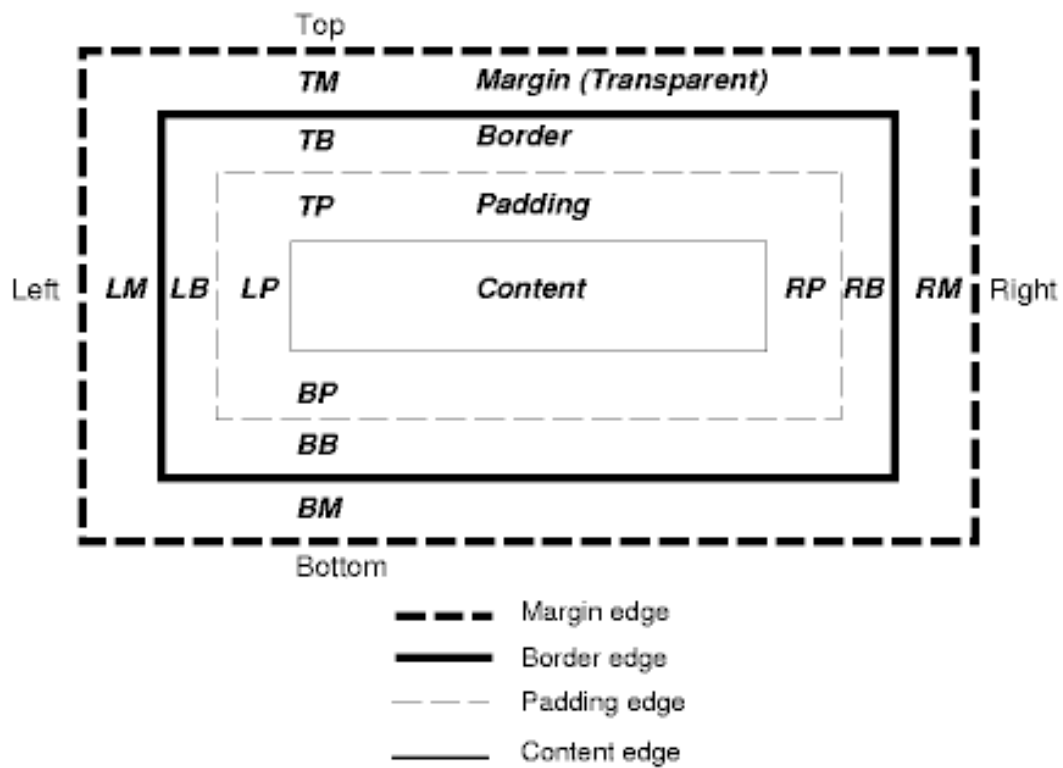
Do not use a hack that relies on a browser's inability to comply with an upcoming CSS specification. You will likely regret it.

Try to only use a hack in a situation where you have a choice of several hacks.

A hard lesson to learn with IE7; and the unfortunate demise of the “*html” hack.

As legacy browsers are phased out, hacking will become a thing of the past. Long live IE7.

CSS Box Model



Great Resources

Books

- Designing with Web Standards, Jeffery Zeldman
- Eric Meyer on CSS, Eric Meyer
- CSS Mastery: Advanced Web Standards Solutions, Andy Budd

One NASA Look and Feel Standards Templates

- <http://portalcss.gsfc.nasa.gov> (* Web Standards to the rescue)

Web Sites

- <http://www.alistapart.com>
- <http://www.positioniseverything.net> (* good info on hacks)
- http://www.456bereastreet.com/lab/developing_with_web_standards/
- <http://www.webstandards.org>
- <http://www.webstandardsgroup.org>
- <http://www.csszengarden.com>
- <http://www.thenoodleincident.com/tutorials/css>
- <http://meyerweb.com/eric/css/>

alistapart.com useful articles (Click on Articles at the top, and browse)

- [12 lessons for those afraid of CSS and Standards](#)
- [Fix your site with the right doctype](#)
- [CSS Design: Going to print](#)
- [CSS Design: Taming lists](#)
- [Sliding doors of CSS part 1](#) - [Sliding doors of CSS part 2](#)
- [Retooling Slashdot part1](#) - [Retooling Slashdot part 2](#)
- [Faux columns](#)
- [Zebra tables](#)
- [CSS sprites: Image slicing's kiss of death](#)
- [Creating liquid layouts with negative margins](#)
- [Accessible pop-up links](#)